John von Neumann Institute for Computing

A High-Performance Parallel Device Simulator for High Electron Mobility Transistors

N. Seoane, A.J. García-Loureiro, K. Kalna, A. Asenov

published in

Parallel Computing:
Current & Future Issues of High-End Computing,
Proceedings of the International Conference ParCo 2005,
G.R. Joubert, W.E. Nagel, F.J. Peters, O. Plata, P. Tirado, E. Zapata (Editors),
John von Neumann Institute for Computing, Jülich,
NIC Series, Vol. **33**, ISBN 3-00-017352-8, pp. 407-414, 2006.

http://www.fz-juelich.de/nic-series/volume33

A High-Performance Parallel Device Simulator for High Electron Mobility Transistors

N. Seoane [a], A. J. García–Loureiro [a], K. Kalna [b] and A. Asenov [b]

[a]Department of Electronics and Computer Engineering, Univ. Santiago de Compostela, 15782 Santiago de Compostela, Spain. e-mail: natalia@dec.usc.es

[b]Device Modelling Group, Dept. Electronics & Electrical Engineering, University of Glasgow, G12 8LT Glasgow, United Kingdom.

Abstract

Three–dimensional simulators are nowadays essential in semiconductor device simulation in order to study fluctuation effects when devices are scaled to gate lengths approaching nanometre dimensions. To take into account these effects it is necessary to perform statistical studies of atomistic simulations, which have a high computational cost, being essential its minimization. In this work we carry out an analysis of the parallel performance of a 3D device simulator for HEMTs based on the drift–diffusion approximation. We also analyse the convenience of reusing the ILU factorisations in order to minimize execution times. Numerical results show superlinear efficiency values up to 32 processors in the resolution of the Poisson equation, and a lowering of the performance with the increase of the number of processors in the solution of the electron continuity equation. The results were obtained in a Cluster HP Integrity Superdome.

1. Introduction

High Electron Mobility Transistors (HEMTs) [1] are being scaled to gate lengths approaching nanometre dimensions. At these scales, the influence of several sources of fluctuations in doping and material composition may significantly degrade the reliability and performance of the devices. In this case 2D models, which neglect the depth of the device, can not be used to take into account these fluctuations effects and have to be replaced by 3D simulations.

To study the impact of fluctuations it is necessary to perform statistical analysis, which increase the computational cost. Standard workstations are not well suited to carry out the large number of simulations required to get statistically significant results keeping a reasonable execution time. To overcome this problem, parallel computers have to be employed in order to speed–up the whole simulation process.

In this work, we describe the design and investigate the parallel performance of a 3D parallel device simulator [2] for HEMTs, based on the drift-diffusion (D–D) approach to the semiconductor transport. The D–D approach constitutes a system of coupled, nonlinear partial differential equations. Finite element methods have been applied to discretise these equations by using tetrahedral elements. Domain decomposition methods have been used to solve the linear systems arising from the linearisation of the D–D equations. The 3D simulator has been developed for multicomputers using a Multiple Instruction Multiple Data strategy (MIMD) under the Single Program Multiple Data paradigm (SPMD) and the Message Passing Interface (MPI) standard library.

The paper is organised as follows. Section 2 briefly presents the mathematical expressions of the D–D transport model. Section 3 describes the numerical techniques used in the simulation process. Results obtained are presented in Section 4 while conclusions are drawn up in Section 5.

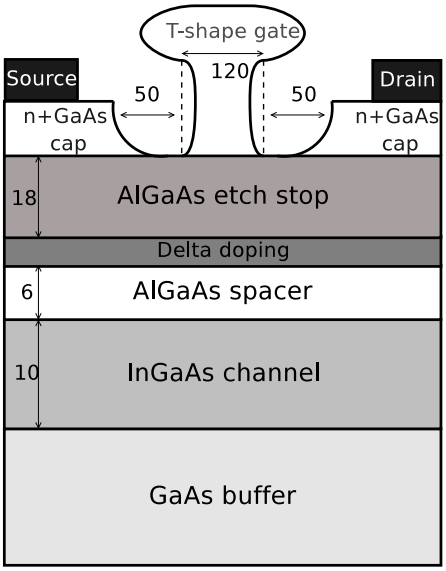

Figure 1. Cross–section of the 120nm gate length pHEMT structure

2. Physical Model

Our 3D parallel device simulator is based on the D–D transport model. The drift–diffusion equations of this model are a system of three coupled, nonlinear equations which describe the relation between the electrostatic potential and the densities of the charge carriers in a semiconductor device. The equations of this model are Poisson equation and electron and hole continuity equations. In stationary state they can be written in the following form:

$$\begin{cases} Find\ (\phi, \phi_n, \phi_p)\ so\ that \\ -div(\epsilon\nabla\phi) + q[n(\phi,\phi_n) - p(\phi,\phi_p) - N_D^+ + N_A^-] = 0 \\ -div(q\mu_n n(\phi,\phi_n)\nabla\phi_n) + qGR(\phi,\phi_n,\phi_p) = 0 \\ -div(q\mu_p p(\phi,\phi_p)\nabla\phi_p) - qGR(\phi,\phi_n,\phi_p) = 0 \\ with\ mixed\ Dirichlet--Neumann\ boundary\ conditions. \end{cases} \quad (1)$$

The unknowns of the problem are ϕ, the electrostatic potential, ϕ_n, the quasi–Fermi level for the electrons and ϕ_p, the quasi–Fermi level for the holes. The electron charge is denoted by q. The mobilities of the electrons and the holes are denoted by μ_n and μ_p respectively and are material dependent. GR is a function which represents the total recombination term. This function may have different expressions depending on the physics taken into account. N_D^+ and N_A^- are the doping effective concentration. The concentration in electrons and holes are n and p.

We have implemented a specific formulation to accelerate the simulation time for HEMTs, due to they are n–type majority carrier devices. Therefore, far from a breakdown we can neglect the hole continuity equation and solve only the Poisson and the electron continuity equation.

3. Three–Dimensional Drift–Diffusion Simulation

The solution scheme is based on the decoupling of the nonlinear Poisson and electron continuity equations in an iterative process. These two equations are discretised using the finite element method (FEM). We have used an unstructured tetrahedral mesh where we have placed more nodes near the

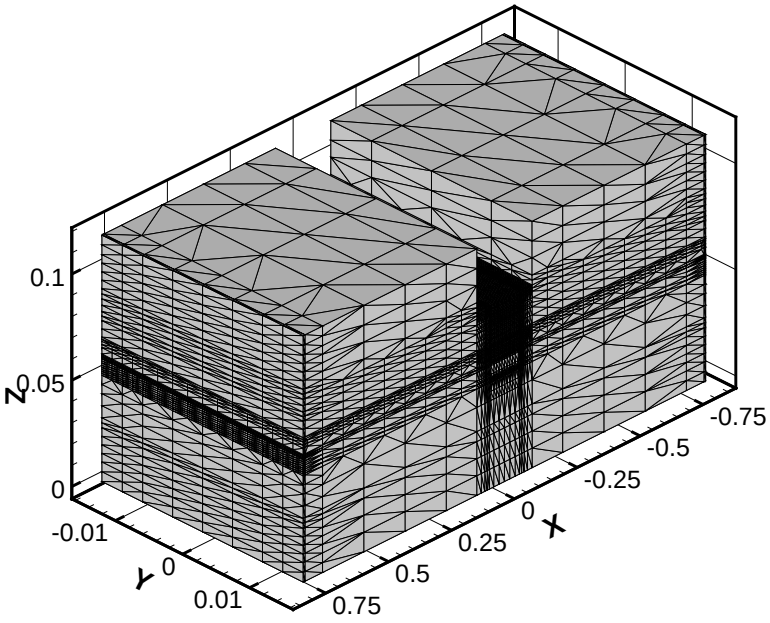

Figure 2. Tetrahedral mesh of the HEMT device generated by the MMG program

interface between different areas of the device, because it is in these areas where we have the greatest gradients of the unknowns of the problem. Then, the METIS program [3] is used to partition the mesh. In this way, the mesh is divided into sub–domains such that we have one for each processor. The same program was subsequently used to relabel the nodes in the sub–domains with the purpose of obtaining a more suitable rearrangement.

The discretisation of the equations leads to two nonlinear systems of algebraic equations. Each of these nonlinear systems is solved by a Newton–Raphson iterative method [4]. Moreover, at every step of the Newton method, a particular linear system has to be solved. The linear system is sparse, badly scaled and ill–conditioned due to the high dynamics of the quantities involved in the simulation and the lack of diagonal dominance in the case of the electron continuity equation [5].

To solve the linear systems of equations we have employed the PSPARSLIB library [6]. This library solves sparse linear systems which are distributed over processors. It uses domain decomposition preconditioners, such as Additive Schwarz, Multicolor SOR and Schur complement methods.

Domain decomposition methods refer to a collection of techniques based on the principle of divide and conquer. If we consider the problem of solving an equation on a domain Ω partitioned in p subdomains Ω_i, such that

$$\Omega = \bigcup_{i=1}^{p} \Omega_i \tag{2}$$

domain decomposition methods attempt to solve the problem on the entire domain by a problem solution on each local subdomain Ω_i.

An analysis of the resolution methods and preconditioning techniques employed in the PSPARSLIB library has been done [7], and the lowest execution times were obtained with the Additive Schwarz method. This algorithm is similar to a block–Jacobi iteration and consists of updating all the new

Table 1
General information about the meshes used in the simulation

Name	Nodes	Tetrahedrons	NNZ	Mesher
S1	26,726	144,608	380,672	MMG
S2	29,012	147,682	398,102	QMG
M1	76,446	433,824	1,116,664	MMG
L1	221,760	1,253,760	3,223,110	MMG

components from the same residual. The basic additive Schwarz iteration would therefore be as follows:

1. Obtain $y_{i,ext}$
2. Compute local residual $r_i = (b - Ax)_i$
3. Solve the local linear system $A_i\delta_i = r_i$
4. Update solution $x_i = x_i + \delta_i$

where $y_{i,ext}$ are the external interface nodes.

To solve the linear system $A_i\delta_i = r_i$ a standard Incomplete LU factorisation with Threshold (ILUT) preconditioner combined with Flexible Generalized Minimal Residual method (FGMRES) for the solver associated with the blocks is used [8]. This is a right–preconditioner variant of the GMRES method that allows the preconditioner to vary at each step. Some zeros in the original matrix may well become nonzeros during the course of ILUT factorisation. The number of the new nonzero elements is indicated with the defined fill–in parameter.

One factor which can affect the convergence of the linear system is the tolerance used for the inner solver. As accuracy increases, the number of outer steps may decrease. However, since the cost of each inner solver increases, this often offsets any gains made from the reduction in the number of outer steps to achieve convergence. It is interesting to observe that the required communication, as well as the overall structure of the routine, is identical with that of matrix–vector products.

4. Numerical Results

The numerical results have been obtained in an HP Superdome Cluster [9] formed by two HP Integrity Superdome servers, each with 64 Itanium2 1.5 GHz, 6 MB cache processors. The main memory of the system is 384 Gbytes and the theoretical peak performance is 768 Gflops.

The 3D device drift–diffusion simulator has been applied to study of a 120nm pHEMT. Simulated characteristics have been compared to data obtained for the 120nm gate length pHEMT designed and fabricated by the Nanoelectronics Research Centre at the University of Glasgow [10]. The schematic cross-section of the simulated device is shown in Figure 1.

The meshing in the 3D simulator is carried out using two programs, the QMG [11] and the MMG [12]. An example of a tetrahedral mesh arising from the MMG program is shown in Figure 2. To accomplish our study we employ four meshes with different size. Their main characteristics are shown in Table 1.

The study has been divided in two sections. In the first one we present, for the Poisson equation in equilibrium, an analysis of the convenience when reusing the ILU factorisations which are used

Table 2
Solving times for Poisson equation in equilibrium reusing the ILU factorisation

Mesh	Processors	$t_{no\_reusing}$(s)	$t_{reusing\_1\_iter}$(s)	$t_{reusing\_2\_iter}$(s)	$t_{only\_a\_first\_ILU}$(s)
S1	1	2250	2200	2174	2150
	2	848	770	783	1013
	4	269	254	248	287
	8	96	80	77	81
	16	38	31	30	32
M1	2	7206	5544	6262	9042
	4	2000	1841	1848	1849
	8	697	655	633	628
	16	214	189	170	181
	32	70	60	57	65
L1	8	5543	4920	6560	4379
	16	1634	1609	1678	1638
	32	591	571	609	746

as preconditioners for Newton iterations and a study of the parallel efficiency. In the second section we show the parallel efficiency of the complete device code.

For this purpose, we have employed the standard definition of the efficiency

$$E(p) = \frac{t_1}{t_p\, p} \tag{3}$$

where t_1 and t_p are the times to execute the workload on a single processor or on p processors respectively.

4.1. Parallel Efficiency of the Poisson Equation in Equilibrium

The first part of this work is related to the evaluation of the ILU factorisations reuse on the parallel performance. ILU factorisations are used as preconditioners for the linear systems with the aim of minimise the execution time. To solve the linear systems we employ the Additive Schwarz domain decomposition method and implement it in each subdomain of the FGMRES solver. FGMRES is preconditioned by the incomplete LU factorisations depending on a particular level of fill–in. In this case, the *fill* parameter used to obtain our results is 70, being $2 \cdot fill$ the maximum number of fill–in elements per row that can be introduced in the structure of outgoing data. Although the matrices change during the Newton iterations, it is possible to reuse same factorisations as an attempt to minimise the cost of solving the linear systems.

The reuse of factorisations slightly improves the performance. Table 2 illustrates that this is more important in the sequential case, where the execution time for Poisson equation always decreases with the increase in the reuse of the same ILU factorisation. However, with the increase of the number of processors employed, reusing always an initial ILU becomes less efficient and we obtain the lowest execution time when we reuse the same ILU one or two iterations.

Figures 3 and 4 show the parallel efficiency for the solution of the Poisson equation in equilibrium for the meshes S1 and M1. Similar results were obtained using the mesh L1. It is also shown the influence of different ILU factorisation reusing conditions for comparative purpose. As we can see we have obtained a surprisingly high superlinearity behaviour. Moreover, in all the studied cases, the parallel efficiency increases with the number of processors employed. The relative increase

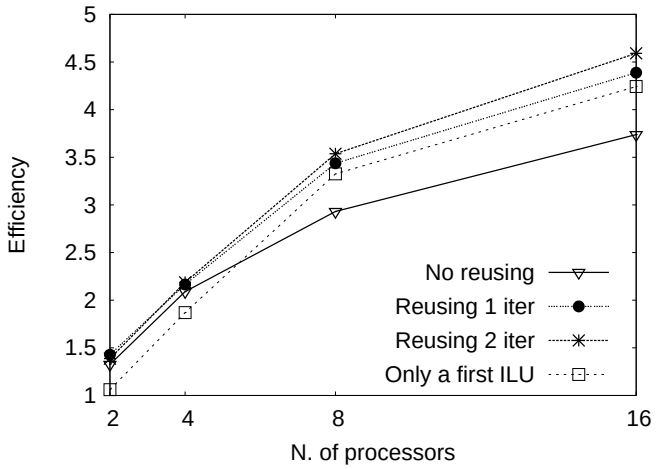

Figure 3. Parallel efficiency for the solution of the Poisson equation in equilibrium using the S1 mesh

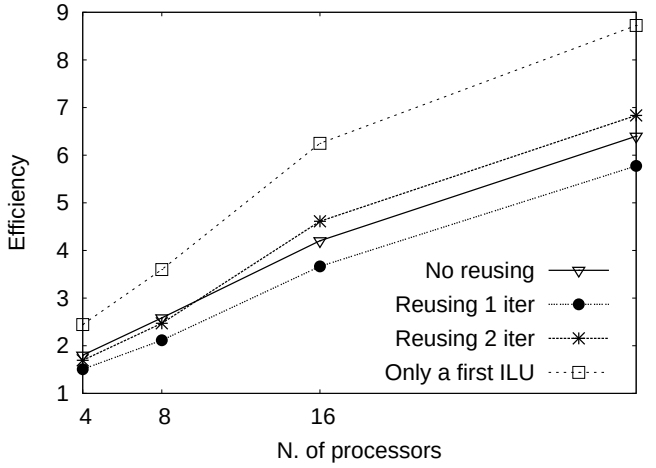

Figure 4. Parallel efficiency for the solution of the Poisson equation in equilibrium using the M1 mesh

is approximately constant, being on average about 1.4 when doubling the number of processors, provided that the size of the subdomains is not too small.

To explain the superlinearity we consider the influence of three factors. First, we take into account cache effects. As the number of processors increase, the number of cache misses decrease. More processors result in a smaller mesh partition size, therefore it fits better in the cache. The higher cache hit rate also results in fewer memory conflicts. The subsequent reduction in memory access times contributes to the parallel efficiency [13]. Second, we consider the influence of the increase of the number of processors in the iterative method, since methods based on domain decomposition are highly parallel [14]. As we stated above, to solve the local nodes within each subdomain we have employed a FGMRES iterative algorithm. Different preconditioning techniques can be applied to this method, we have tried both the PGMRES iterative method and the ILU preconditioner. PGMRES procedure is a simple version of the ILUT preconditioned GMRES algorithm. Although we have found a slight increase in the number of FGMRES iterations when we increase the number

Table 3
Solving time and efficiency for the solution of the Poisson equation in equilibrium and the complete simulation for the S2 mesh

Processors	Poisson time(s)	Poisson efficiency	Complete sim. time(s)	Complete sim. efficiency
1	2698		6579	
2	1099	1.228	2839	1.158
4	428	1.576	1593	1.032
6	297	1.511	939	1.167
16	65	2.580	634	0.648

of processors employed, the cost of each iteration noticeably decreases. And finally, we have to take into account ILU factorisations. The reduction in the size of the matrix is not linear with the increase of the number of processors, therefore the lowering in the factorisation time is higher than the increase in the number of processors.

4.2. Parallel Efficiency of the Complete 3D Simulator

The efficiency analysis of the 3D parallel simulator has been divided in two parts. First, we have solved the Poisson equation in equilibrium. Then, we have obtained the complete simulation time for one point on the I-V curve. In this case the contribution of the Poisson and electron continuity equations are considered. Due to the electron continuity equation properties it is necessary to increase the fill–in in order to achieve the convergence of the system. Therefore the *fill* parameter used to obtain our performance results is 700.

The obtained results for the S2 mesh are summarised in Table 3. The second two columns in this table illustrate time and efficiency for the solution of the Poisson equation in equilibrium. This task is very well parallelizable which can be seen from the superlinear efficiency for up to 16 processors used in this investigation. The execution times are higher than the ones obtained in the previous section because of the different value of fill–in employed.

The complete simulation time and efficiency for a one point on I-V characteristics are also shown in the last two columns of Table 3. The behaviour of the complete simulation for the one point is different because of the influence of the electron continuity equation. In this case, the efficiency drops with the increase of the number of processors. In domain decomposition methods, the partition of the mesh into a higher number of subdomains leads to a reduction of the internal nodes, which causes an increase both in the communications and in the computational effort, mainly due to the higher number of iterations of the inner solver and therefore in the matrix–by–vector multiplications, which are usually the most time consuming part of the iterative solver. This increase in the number of iterations is much more noticeable when we are solving the electron continuity equation, causing a decrease in the parallel performance.

5. Conclusions

3D parallel simulations are essential tools in order to study effects of fluctuations, both in doping and in material composition, when semiconductor devices are scaled into deep submicron dimensions. In this work we have developed a high performance parallel devices simulator for High Electron Mobility Transistors (HEMTs).

The objectives of this paper were twofold, first to analyse the convenience of reusing the ILU factorisations for the Poisson equation used as preconditioners in order to minimize execution times. Second, to study the parallel performance of a 3D parallel semiconductor device simulator, based on the drift–diffusion approximation to the semiconductor transport.

The obtained results indicate that the reuse of the ILU factorisations slightly improves the performance, obtaining the lowest execution times when we reuse the factorisations one or two iterations. With respect to the parallel efficiency study, the resolution of the electron continuity equation is the bottle–neck of the simulation, limiting the scalability and performance of the simulation. On the other hand, the solution of the Poisson equation obtains high parallel efficiency, presenting superlinear behaviour in all the studied cases.

Open questions remain in this study. The results have shown that the main limitation to the performance of the 3D simulator is the solution of the electron continuity equation, so that it should be very interesting to apply a more suitable resolution method and preconditioning technique to solve the linear systems associated with the electron continuity equation.

Acknowledgements.

This work was partly supported by the Spanish Government (CICYT) under the project TIN2004-07797-C02. It has also been performed under the Project HPC–EUROPA (RII3–CT–2003–506079), with the support of the European Community – Research Infrastructure Action under the FP6 Structuring the European Research Area Programme. We are particular grateful to CESGA (Galician Supercomputing Center) and Carlos Fernández for providing access to the HP Superdome system.

References

[1] P. Roblin and H. Rohdin, "High–speed heterostructure devices", Cambridge University Press (2002).
[2] A. García Loureiro, K. Kalna and A. Asenov, "3D Parallel Simulations of Fluctuation Effects in pHEMTs", *J. Comput. Electron.* **2**, 369-373 (2003).
[3] G. Karypis and V. Kumar, "METIS: A software package for partitioning unstructured graphs", University of Minnesota, 1997.
[4] R.E. Bank and D.J. Rose, "Global approximate Newton Methods", *Numerische Matematik*, **37**, 279-295 (1981).
[5] S. Rollin, O. Schenk and A. Gupta, "The effects of unsymmetric matrix permutations and scalings in semiconductor device and circuit simulation", *IEEE Trans. CAD Integ. Circ. Systems*, **23** (2004).
[6] Y. Saad, Gen-Ching Lo, and S. Kuznetsov, "PSPARLIB users manual: A portable library of parallel sparse iterative solvers", Technical report, University of Minnesota, 1997.
[7] N. Seoane and A. García Loureiro, "Analysis of Parallel Numerical Libraries to Solve the 3D Electron Continuity Equation", *Lecture Notes in Computer Science*, **3036**, 590-593 (2004).
[8] Y. Saad, "Iterative methods for sparse linear systems", PWS Publishing Co. (1996).
[9] Galician Supercomputing Center, http://www.cesga.es
[10] K. Kalna, A. Asenov, K. Elgaid and I. Thayne, *Solid State Electron*, **46**, 631 (2002).
[11] S. A. Vavasis, "QMG 1.1 Reference Manual", Computer Science Department, Cornell University, 1996.
[12] M. Aldegunde, Juan J. Pombo, A. García Loureiro, "Octree–based mesh generation for the simulation of semiconductor devices", XX Conference on Design of Circuits and Integrated Systems (DCIS), 2005.
[13] "Itanium processor family performance tuning guide", Technical paper.
[14] A. Quarteroni and A. Valli, "Domain Decomposition Methods for Partial Differential Equations", Oxford University Press (1999).